

Cryptography CS 555

Lecture 18



Department of Computer Sciences
Purdue University

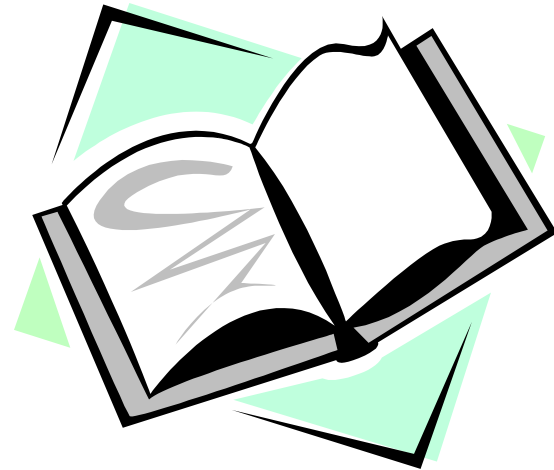
Lecture Outline

- Challenge-response
- Zero-knowledge



Recommended Reading

- HAC: Chapter 10 for authentication protocols



Review: Entity Authentication Protocols

- Password/PIN
- One-time password

Challenge-Response Protocols

- Goal: one entity authenticates to other entity proving the knowledge of a secret by answering a 'challenge'
- Time-variant parameters used to prevent replay, interleaving attacks, provide uniqueness and timeliness : nonce (used only once)
- Three types:
 - Random numbers
 - Sequences
 - Timestamp

Types of Challenges

- **Random numbers:**
 - pseudo-random numbers that are unpredictable to an adversary;
 - vulnerable to birthday attacks, use larger sample;
 - must maintain state;
 - do not prevent interleaving attacks (parallel sessions)
- **Sequences:**
 - serial number or counters;
 - long-term state information must be maintained by both parties+ synchronization
- **Timestamp:**
 - provides timeliness and detects forced delays;
 - requires synchronized clocks.

Challenge-response based on symmetric-key encryption

- Unilateral authentication, timestamp-based
 - A to B: $E_K(t_A, B)$
- Unilateral authentication, random-number-based
 - B to A: r_B
 - A to B: $E_K(r_B, B)$
- Mutual authentication, using random numbers
 - B to A: r_B
 - A to B: $E_K(r_A, r_B, B)$
 - B to A: $E_K(r_B, r_A)$

Challenge-Response Protocols Using Digital Signatures

- unilateral authentication with timestamp
A → B: cert_A, t_A, B, S_A(t_A, B)
- unilateral authentication with random numbers
A ← B: r_B
A → B: cert_A, B, S_A(r_B, B)
- mutual authentication with random numbers
A ← B: r_B
A → B: cert_A, r_A, B, S_A(r_A, r_B, B)
A ← B: cert_B, A, S_B(r_B, r_A, A)

Zero Knowledge Proofs

- A kind of interactive proof system
- Involves a prover and a verifier
- Proving without revealing any other information

Two Kinds of Zero-Knowledge Proofs

- ZK proof of a statement
 - convincing the verifier that a statement is true without yielding any other information
 - example of a statement, a propositional formula is satisfiable
- ZK proof of knowledge of a secret
 - convincing the verifier that one knows a secret, e.g., one knows the square root modulo $N=pq$

Properties Zero-Knowledge Proofs

- Properties of ZK Proofs:
 - completeness
 - honest prover who knows the secret convinces the verifier with overwhelming probability
 - soundness
 - no one who doesn't know the secret can convince the verifier with nonnegligible probability
 - zero knowledge
 - the proof does not leak any additional information
- How to formalize soundness and ZK?

Fiat-Shamir ID protocol (ZK Proof of knowledge of square root modulo n)

- System parameter: $n=pq,$
- Public identity: $v \quad v = s^2 \text{ mod } n$
- Private authenticator: s
- Protocol (repeat t times)
 1. A: picks random r in Z_n^* , sends $x=r^2 \text{ mod } n$ to B
 2. B checks $x \neq 0$ and sends random c in $\{0,1\}$ to A
 3. A sends y to B, where If $c=0$, $y=r$, else $y=rs \text{ mod } n$.
 4. B accept if $y^2 \equiv xv^c \text{ mod } n$

Observations on the Protocol

- Multiple rounds
- Each round consists of 3 steps
 - commit
 - challenge
 - respond
- If challenge can be predicted, then cheating is possible.
 - cannot convince a third party (even if the party is online)
- If respond to more than one challenge with one commit, then the secret is revealed.

Formalizing Soundness

- If a prover A can convince a verifier, then a knowledge extractor exists
 - a polynomial algorithm that given A can output the secret

Formalizing ZK property

- A simulator exists
 - taking what the verifier knows before the proof, can generate a communication transcript that is indistinguishable from one generated during ZK proofs
- Three kinds of indistinguishability
 - perfect (information theoretic)
 - statistical
 - computational

Guillou-Quisquater Id protocol (ZK Proof of knowledge of e'th root)

- System parameter: $n=pq$
- Public identity: $v \quad v = s^e \text{ mod } n$
- Private authenticator: s
- Protocol: Repeat t times
 1. A: picks random r in Z_n^* , sends $x=r^e \text{ mod } n$
 2. B: checks $x \neq 0$ and sends random challenge c in $[1..e]$ to A;
 3. A sends $y=r \cdot s^c \text{ mod } n$
 4. B accpets if $x=(y^e v^{-c} \text{ mod } n)$

Security of GQ Id protocol

- Probability of successful cheating in one round: $1/e$
- Security assumption required: factor n
- Parameter choosing in practice:
 - one round with large (k-bit) e

Schnorr Id protocol (ZK Proof of Discrete Log)

- System parameter: p, q, g
 - $q \mid (p-1)$ and g is an order q element in Z_p^*
- Public identity: v
- Private authenticator: s $v = g^{-s} \bmod p$
- Protocol
 1. A: picks random r in $[1..q]$, sends $x = g^r \bmod p$,
 2. B: sends random challenge c in $[1..2^t]$
 3. A: sends $y = sc + r \bmod q$
 4. B: accepts if $x = (g^y v^c \bmod p)$

Security of Schnorr id protocol

- probability of forgery: $1/2^t$
- soundness:
- ZK property: not ZK if $2^t > \log n$ is used

Converting Interactive ZK to Non-interactive ZK

- The only interactive role played by the verifier is to generate random challenges
 - challenges not predictable by the prover
- The same thing can be done using one-way hash functions

Interactive ZK Implies Signatures

- Given a message M , replace the random challenge of the verifier by the one-way hash $c=h(x||M)$

Next Lecture...

- Authentication & Key Agreement Protocols
- Network Authentication services: Kerberos

